REMARKS

This is a response to the final Office Action mailed January 21, 2009.

Claims 20, 27, 36 and 47 are amended. No new matter is presented.

Rejection under 35 U.S.C. §112, second paragraph

Claims 20, 27, 36 and 47 were rejected under 35 U.S.C. §112, second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Claims 20, 27, 36 and 47 have been amended herein to refer to "transcoded media content" instead of "transformed media content." Withdrawal of the rejection is therefore respectfully requested.

Rejection under Tuli in view of Davis et al.

Claims 1, 5, 6, 9, 13, 21-24, 27, 28, 30-32, 37-40, 45-50, 53, 54, 67, 69, 70, 76 and 81 were rejected under 35 U.S.C. §103(a) a being unpatentable over Tuli (US 7,068,381) in view of Davis et al. (US 6,643,696). Applicants believe that claim 83 is included in this rejection as well.

Claim 1

Claim 1 is patentable for at least the following reasons.

Regarding "said mark-up language description includes one or more source files which describe a behavior of said particular content on a user interface of said user device based on user interactions with the particular content via the user interface," the Office asserts at par. 92 that "the user is enabled to interact with the browser window" due to the use of "virtual clicks." In particular, Tuli sends bitmap or raster image data ("content") to a user device which is used to display an image, and the user uses a pointing device to click on the image. This event is sent to a server and processed to provide a "virtual click" to a "virtual browser" (col. 2, line 64 to col. 3, line 6). *However, the bitmap or raster image data is only statically displayed and does not respond to user interactions. Further, there is no description of a behavior of content to user interactions with the content via a user interface, based on code transmitted to a user device.*

Arguably, with Tuli, the user interacts with the user device/browser window; however, this involves a behavior of the user device, such as in sending a click to a server – not a behavior of the displayed content.

Regarding "compiling ... to create executable code," the Office Action at par. 93 states: "For the information processed to work at all, some sort of code must be provided to the user's device to work properly." Although Tuli does not provide details, it is well know to provide locally stored code which is executed by a CPU. That is, a person of ordinary sill in the art would conclude that the code which is executed by Tuli to render an image based on the received image data is *already at the user's device. There is no mention by Tuli of receiving any such code from the server to process the image data.*

Further, Applicants have stated that they do not believe Tuli's use of image data (raster, bitmap or JPEG data) constitutes executable code. It is helpful to understand how image data such as raster or bitmap data is processed. As explained in the attached definitions from Microsoft Computer Dictionary (2002), a bitmap is "a *data structure* in memory that represents information in the form of a collection of individual bits." See also the definition of "bitmapped graphics." Further, a "data structure" (example definition attached) is an organization scheme, such as a record or array, that can be applied to data *to facilitate interpreting the data or performing operations on it*." *Thus, the image data that Tuli's user device obtains from a server is simply a data structure of bits which is interpreted or has operations performed on it, e.g., to display an image. Under any common definition which would be used by those skilled in the art, the image data of Tuli is a data structure and not executable code.*

Further, Applicants have previously pointed out that an "executable program" is defined as "a program that can be run," and "code" is defined as "program instructions" (example definition attached).

For the simple case of monochrome/black and white data (Tuli, col. 1, line 33), those skilled in the art recognize that the bits can represent a black or white pixel, and a position of the pixel on a screen, and that the CPU of Tuli will execute code to read each bit and cause the display to output a black or white pixel at the specified position on the screen.

Accordingly, the inevitable conclusion is that *the image data of Tuli's user device is not executable code as set forth in claim 1 because the image data is read by executed code.* While claims can be interpreted during examination using the broadest reasonable interpretation, the interpretation must also be consistent with the interpretation that those skilled in the art would reach. In re Cortright, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999).

Applicants respectfully submit that the Examiner's claim interpretation does not meet this requirement.

Withdrawal of the rejection to claim 1 is therefore respectfully requested.

Claim 5

Par. 14 and 96 of the Office Action cite Harrington (US2002/0156909) as describing a common usage of providing ActionScript within a Flash player within a browser. However, even if this is true, Tuli teaches against the use of ActionScript within a Flash player within a browser because Tuli seeks to avoid the need for a mini-browser in the client device which requires a powerful microprocessor, and instead provides a browser translator or "virtual browser" in a server (col. 1, line 66 to col. 2, line 21). Tuli's client device is limited to decompressing and displaying bitmap or raster images, scrolling through images and providing virtual clicks (col. 4, lines 25-33). *If Tuli had a full featured browser which could process ActionScript within a Flash player, it would not need to translate web pages to bit map data before sending it to the client device.*

Withdrawal of the rejection to claim 5 is therefore respectfully requested.

Claim 81

Par. 41 of the Office Action states that Tuli at col. 4, lines 18-22 refers to using JPEG to compress a color image which is provided to the remote user device. However, there is no mention that an object which identifies the format is *provided via a user interface*.

Withdrawal of the rejection to claim 81 is therefore respectfully requested.

Rejection under Tuli and Davis et al. in view of Rubin et al.

Claims 4, 7, 36, 52, 55-57, 60, 62, 64, 65, 73, 77, 78, 80 and 82 were rejected under 35 U.S.C. §103(a) as being unpatentable over Tuli and Davis in view of Rubin et al. (US 6,701,522).

Claim 4

Par. 44 of the Office Action asserts that one of ordinary skill would have been motivated to combine Rubin with Tuli because plug-ins are auxiliary programs added to web browsers that provide them with new functionality (Rubin, col. 7, lines 21-23). However, Tuli specifically seeks to avoid the need for a mini-browser which requires a powerful microprocessor (col. 1, line 66 to col. 2, line 4) and does this by moving the browser functionality to the server as a virtual browser (col. 2, lines 9-24). Accordingly, a person of ordinary skill in the art would see that the user device of Tuli does not have a browser functionality which can support a plug-in program.

Withdrawal of the rejection to claim 4 is therefore respectfully requested.

Claim 52

Par. 47 cites col. 2, lines 5-13 of Tuli as providing an indication in a request from a client which identifies a type of a rendering entity of the client from a group of rendering entities. However, this passage only indicates that the web server which receives the request can obtain HTML or Java web pages and convert it to bitmap or raster data for use by the client. There is no mention of any concern with the client identifying a type of a rendering entity in a request to a server.

Withdrawal of the rejection to claim 52 is therefore respectfully requested.

Claim 60

Par. 51 cites Tuli at col. 4, lines 16-22 as providing an animation as claimed. However the cited passage only refers to the use of JPEG color images. *JPEG is a standard for photographs, not animations.*

Withdrawal of the rejection to claim 60 is therefore respectfully requested.

Claim 62

Par. 52 asserts that it would be obvious to modify Tuli to use .SWF file because .SWF files are known. However, this is not the standard for determining obviousness. There must be some teaching, motivation or other reason to try the proposed combination. *The Office has only asserted that it would be obvious to make the proposed combination because the element of a .SWF file is known per se.* Applicants have demonstrated how Tuli is directed to a limited function user device which moves the browser functionality to the server as a virtual browser and which is limited to decompressing and displaying bitmap or raster images, scrolling through images and providing virtual clicks (col. 4, lines 25-33). It is clear that if Tuli had a full featured browser which could process a .SWF file, it would not need to translate web pages to bit map data before sending it to the client device. A person of ordinary skill would therefore not be led to make the proposed combination.

Withdrawal of the rejection to claim 62 is therefore respectfully requested.

Rejection under Tuli and Davis et al. in view of Harrington

Claims 8, 10 and 51 were rejected under 35 U.S.C. §103(a) as being unpatentable over Tuli and Davis in view of Harrington (US 2002/0156909).

Claims 8 and 51

Regarding claim 8, and "accessing media content comprising at least one of audio, video and a movie," the Office Action at par. 61 does not address this feature. It only indicates that Tuli retrieve HTML data. However, this is not at least one of audio, video and a movie. In fact, Tuli's user device can only handle static images (col. 4, lines 25-34).

The Office Action at par. 63 addresses this feature in regard to claim 51, citing Harrington (US2002/0156909). Harrington provides a method for controlling a Flash presentation on a client device. The Office asserts it would be obvious to implement a Flash method with Harrington (Tuli?) to enable the displaying of movie content because of the common usage of Flash players in web browsing environments. However, as mentioned, Tuli seeks to avoid the need for a mini-browser in the client device because it requires a powerful microprocessor, and instead provides a browser translator or "virtual browser" in a server (col. 1, line 66 to col. 2, line 21). The client device is limited to decompressing and displaying bitmap or raster images, scrolling through images and providing virtual clicks (col. 4, lines 25-33). Accordingly, it would not be obvious to modify Tuli's user device to process media content comprising at least one of audio, video and a movie

Withdrawal of the rejection to claims 8 and 51 is therefore respectfully requested.

Claim 10

In claim 10, a media file is transcoded. The Office at par. 97 and 98 refers to Tuli's dividing an image into sections after a bitmap or raster is created, as transcoding. This is done for purposes of display priority, so that one section will be decompressed and displayed before other sections (col. 2, lines 42-49). However, there is no mention that this involves transcoding (e.g., converting a media file or object from one format to another, such as from QuickTime to MPEG - example definition attached). The Office states (par. 98) that content "is sent to a program for division and changed appropriately so that the user's display can display the content in the appropriate manner." However, *Tuli does not mention any "program" which is used for division or "changing" the content.* All Tuli states is that the bitmap or raster image data is divided into sections before it is compressed and transmitted to the user. A person skilled in the art would clearly interpret this to mean that different portions of the bitmap or raster image data can be defined, and that the different portions are compressed in an order according to their priority. In any case, the data must be processed in some order, and Tuli merely defines a preferred order. However, there is simply no mention of transcoding according to its commonly accepted definition. *Instead, the data remains as bitmap or raster image data.*

Further, *claim 10 refers to a media file being transcoded*, where (per claim 8) the media file contains media content comprising at least one of audio, video and a movie. Tuli provides no mention of transcoding such a media file as claimed.

Rejection under Tuli and Harrington in view of Rubin et al.

Claims 28, 30-33, 41-44, 58 and 61 were rejected under 35 U.S.C. §103(a) as being unpatentable over Tuli in view of Harrington and further in view of Rubin.

These claims are patentable at least for the above-mentioned reasons.

Rejection under Tuli and Davis et al. in view of Russell

Claim 11 was rejected under 35 U.S.C. §103(a) as being unpatentable over Tuli and Davis in view of Russell (US 2002/0069420).

This claim is patentable at least by virtue of its dependence on claim 1 which is patentable at least for the above-mentioned reasons.

Rejection under Tuli in view of Wagner

Claims 14-17, 19, 20 and 74-75 were rejected under 35 U.S.C. §103(a) as being unpatentable over Tuli in view of Wagner (US 6,085,224).

These claims are patentable at least for the above-mentioned reasons.

Rejection under Tuli and Rubin et al. in view of Davis et al.

Claims 68 and 79 were rejected under 35 U.S.C. §103(a) as being unpatentable over Tuli and Rubin in view of Davis.

These claims are patentable at least for the above-mentioned reasons.

Conclusion

In view of the above, each of the pending claims is believed to be in condition for immediate allowance. The Examiner is therefore respectfully requested to pass this application on to an early issue.

The Examiner's prompt attention to this matter is greatly appreciated. Should further questions remain, the Examiner is invited to contact the undersigned attorney by telephone.

The Commissioner is authorized to charge any underpayment or credit any overpayment to Deposit Account No. 501826 for any matter in connection with this response, including any fee for extension of time, which may be required.

Respectfully submitted,

Date: April 6, 2009

By: /Ralph F. Hoppin/
Ralph F. Hoppin
Reg. No. 38,494

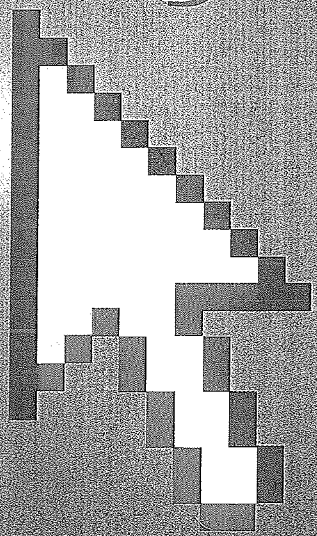
VIERRA MAGEN MARCUS & DENIRO LLP
575 Market Street, Suite 2500
San Francisco, California 94105-4206
Telephone: (415) 369-9660, x214
Facsimile: (415) 369-9665

Microsoft®

OVER
10,000
ENTRIES

Microsoft® Computer Dictionary

Fifth Edition



- *Fully updated with the latest technologies, terms, and acronyms*
- *Easy to read, expertly illustrated*
- *Definitive coverage of hardware, software, the Internet, and more!*

ware routine spe-
problems.

us.

tates, such as posi-
ansfer and process-
a signal in which
on and off, true and
ee also nonreturn to
onics, pertaining to or
two types of charge

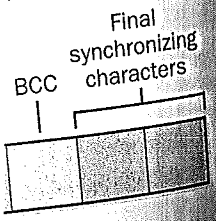
stem.

characteristic of a sys-
e states, such as on and

hat has only two stable
m must be initiated from
cuit is capable of storing

lip-flop.

achronous communica-
ons standard developed by
re encoded in either ASCII
of any length and are sent in
preceded by a message
nous transmission, in which
ed by a specific time interval,
d followed by special charac-
nd receiving machines to syn-
nd ETX are control characters
end of the message text; BCC
verify the accuracy of trans-
. *Also called:* BSC.



bit *n.* Short for **bi**nary digi**t**. The smallest unit of information handled by a computer. One bit expresses a 1 or a 0 in a binary numeral, or a true or false logical condition, and is represented physically by an element such as a high or low voltage at one point in a circuit or a small spot on a disk magnetized one way or the other. A single bit conveys little information a human would consider meaningful. A group of 8 bits, however, makes up a byte, which can be used to represent many types of information, such as a letter of the alphabet, a decimal digit, or other character. *See also* ASCII, binary^1, byte.

bit block *n.* In computer graphics and display, a rectangular group of pixels treated as a unit. Bit blocks are so named because they are, literally, blocks of bits describing the pixels' display characteristics, such as color and intensity. Programmers use bit blocks and a technique called bit block transfer (bitblt) to display images rapidly on the screen and to animate them. *See also* bit block transfer.

bit block transfer *n.* In graphics display and animation, a programming technique that manipulates blocks of bits in memory that represent the color and other attributes of a rectangular block of pixels forming a screen image. The image described can range in size from a cursor to a cartoon. Such a bit block is moved through a computer's video RAM as a unit so that its pixels can be rapidly displayed in a desired location on the screen. The bits can also be altered; for example, light and dark portions of an image can be reversed. Successive displays can thus be used to change the appearance of an image or to move it around on the screen. Some computers contain special graphics hardware for manipulating bit blocks on the screen independently of the contents of the rest of the screen. This speeds the animation of small shapes, because a program need not constantly compare and redraw the background around the moving shape. *Also called:* bitblt. *See also* sprite.

bitblt *n. See* bit block transfer.

bit bucket *n.* An imaginary location into which data can be discarded. A bit bucket is a null input/output device from which no data is read and to which data can be written without effect. The NUL device recognized by MS-DOS is a bit bucket. A directory listing, for example, simply disappears when sent to NUL.

bit data type *n.* In an Access project, a data type that stores either a 1 or 0 value. Integer values other than 1 or 0 are accepted, but are always interpreted as 1.

bit density *n.* A measure of the amount of information per unit of linear distance or surface area in a storage medium or per unit of time in a communications pipeline.

bit depth *n.* The number of bits per pixel allocated for storing indexed color information in a graphics file.

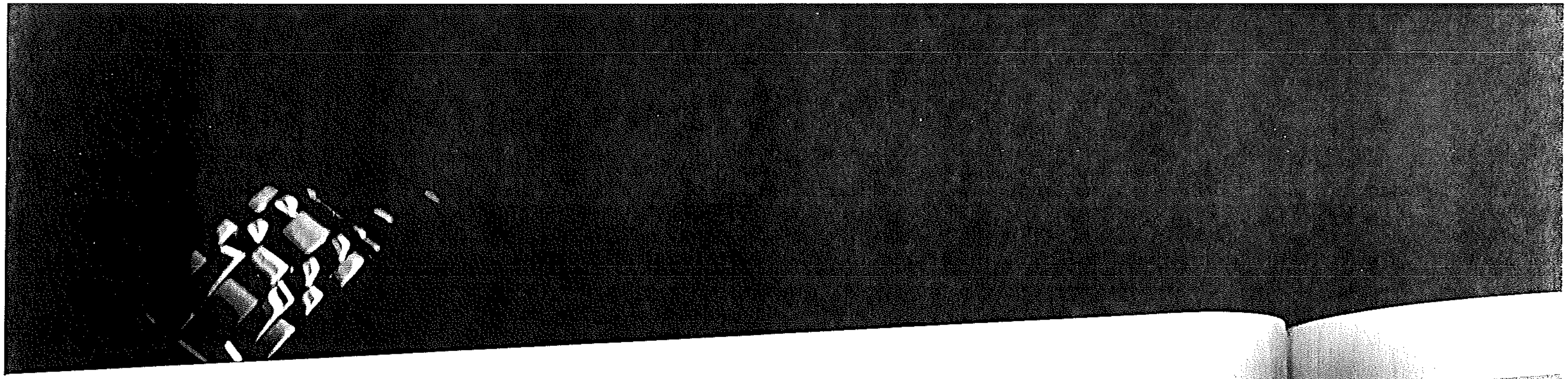
bit flipping *n.* A process of inverting bits—changing 1s to 0s and vice versa. For example, in a graphics program, to invert a black-and-white bitmapped image (to change black to white and vice versa), the program could simply flip the bits that compose the bit map.

bit image *n.* A sequential collection of bits that represents in memory an image to be displayed on the screen, particularly in systems having a graphical user interface. Each bit in a bit image corresponds to one pixel (dot) on the screen. The screen itself, for example, represents a single bit image; similarly, the dot patterns for all the characters in a font represent a bit image of the font. In a black-and-white display each pixel is either white or black, so it can be represented by a single bit. The "pattern" of 0s and 1s in the bit image then determines the pattern of white and black dots forming an image on the screen. In a color display the corresponding description of on-screen bits is called a pixel image because more than one bit is needed to represent each pixel. *See also* bitmap, pixel image.

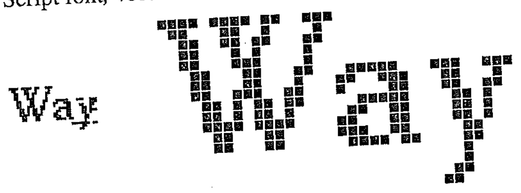
bit manipulation *n.* An action intended to change only one or more individual bits within a byte or word. Manipulation of the entire byte or word is much more common and generally simpler. *See also* mask.

bitmap *n.* A data structure in memory that represents information in the form of a collection of individual bits. A bit map is used to represent a bit image. Another use of a bit map in some systems is the representation of the blocks of storage on a disk, indicating whether each block is free (0) or in use (1). *See also* bit image, pixel image.

bitmapped font *n.* A set of characters in a particular size and style in which each character is described as a unique bit map (pattern of dots). Macintosh screen fonts are examples of bitmapped fonts. See the illustration. *See also*



downloadable font, outline font, TrueType. *Compare* PostScript font, vector font.



Bitmapped font. *Each character is composed of a pattern of dots.*

bitmapped graphics *n.* Computer graphics represented as arrays of bits in memory that represent the attributes of the individual pixels in an image (one bit per pixel in a black-and-white display, multiple bits per pixel in a color or gray-scale display). Bitmapped graphics are typical of paint programs, which treat images as collections of dots rather than as shapes. *See also* bit image, bit map, pixel image. *Compare* object-oriented graphics.

bit mask *n.* A value used with bit-wise operators (And, Eqv, Imp, Not, Or, and Xor) to test, set, or reset the state of individual bits in a bit-wise field value.

BITNET *n.* Acronym for **B**ecause **I**t's **T**ime **Net**work. A WAN (wide area network) founded in 1981 and operated by the Corporation for Research and Educational Networking (CREN) in Washington, D.C. Now defunct, BITNET provided e-mail and file transfer services between mainframe computers at educational and research institutions in North America, Europe, and Japan. BITNET used the IBM Network Job Entry (NJE) protocol rather than TCP/IP, but it could exchange e-mail with the Internet. The listserv software for maintaining mailing lists was originated on BITNET.

bit. newsgroups *n.* A hierarchy of Internet newsgroups that mirror the content of some BITNET mailing lists. *See also* BITNET.

bit-oriented protocol *n.* A communications protocol in which data is transmitted as a steady stream of bits rather than as a string of characters. Because the bits transmitted have no inherent meaning in terms of a particular character set (such as ASCII), a bit-oriented protocol uses special sequences of bits rather than reserved characters for control purposes. The HDLC (high-level data link control) defined by ISO is a bit-oriented protocol. *Compare* byte-oriented protocol.

bit parallel *adj.* Transmitting simultaneously all bits in a set (such as a byte) over separate wires in a cable. *See also* parallel transmission.

bit pattern *n.* **1.** A combination of bits, often used to indicate the possible unique combinations of a specific number of bits. For example, a 3-bit pattern allows 8 possible combinations and an 8-bit pattern allows 256 combinations. **2.** A pattern of black and white pixels in a computer system capable of supporting bitmapped graphics. *See also* pixel.

bitplane *n.* **1.** One of a set of bit maps that collectively make up a color image. Each bit plane contains the values for one bit of the set of bits that describe a pixel. One bit plane allows two colors (usually black and white) to be represented; two bit planes, four colors; three bit planes, eight colors; and so on. These sections of memory are called bit planes because they are treated as if they were separate layers that stack one upon another to form the complete image. By contrast, in a chunky pixel image, the bits describing a given pixel are stored contiguously within the same byte. The use of bit planes to represent colors is often associated with the use of a color look-up table, or color map, which is used to assign colors to particular bit patterns. Bit planes are used in the EGA and VGA in 16-color graphics modes; the four planes correspond to the 4 bits of the IRGB code. *See also* color look-up table, color map, EGA, IRGB, layering, VGA. *Compare* color bits. **2.** Rarely, one level of a set of superimposed images (such as circuit diagrams) to be displayed on the screen.

bit rate *n.* **1.** The speed at which binary digits are transmitted. *See also* transfer rate. **2.** The streaming speed of digital content on a network. Bit rate is usually measured in kilobits per second (Kbps).

bit serial *n.* The transmission of bits in a byte one after another over a single wire. *See also* serial transmission.

bit slice microprocessor *n.* A building block for microprocessors that are custom-developed for specialized uses. These chips can be programmed to handle the same tasks as other CPUs but they operate on short units of information, such as 2 or 4 bits. They are combined into processors that handle the longer words.

bits per inch *n.* A measure of data storage capacity; the number of bits that fit into an inch of space on a disk or a tape. On a disk, bits per inch are measured based on inches of circumference of a given track. *Acronym:* BPI. *See also* packing density.

bits per pixel *n.* Also
The term refers to the
used to store and displ
The number of bits pe
available to an image.

bits per second *n.* S

bit stream *n.* **1.** A se
flow of information t
2. In synchronous co
data in which charact
one another by the re
ers, such as start and

bit stuffing *n.* The
stream of transmitte
that a special sequer
locations. For exam
communications pro
only at the beginnin
so bit stuffing is use
stream whenever fi
bits are removed by
to its original form

bit transfer rate

bit twiddler *n.* Sl
particularly one w
guage. *See also* ha

BIX *n.* Acronym f
online service orig
and operated by I
BIX offers e-mail
relating to hardw

.biz *n.* One of se
approved in 2000
Names and Num
business-related

biz. news grou
the biz. hierarch
groups are devo
Unlike most oth
permit users to
material. *See a*
hierarchy.

;pecification *n.*

curs when more
essed. *See also* bps.

llows as it travels

lues plotted on a

ork performed by
nanipulation of data
lt. *Acronym:* DP.
ocessing, EDP, elec-
alized processing,
processing.

ssociation *n. See*

to a slide projector,
ut of a computer onto

ensuring the preserva-
. *See also* data integrity.

circuit or communica-
usually measured in

converting raw data to
oothing, ordering, or

memory or auxiliary
d by a program.

lated information made
be treated as a unit in data
, a modem. *See also*

ngle file by more than one
g can be done by physi-
e computer to another, or
; and computer-to-computer

data signal *n.* The information transmitted over a line or circuit. It consists of binary digits and can include actual information or messages and other elements such as control characters or error-checking codes.

data sink *n.* **1.** Any recording medium where data can be stored until needed. **2.** In communications, the portion of a Data Terminal Equipment (DTE) device that receives transmitted data.

data source *n.* **1.** The originator of computer data, frequently an analog or digital data collection device. **2.** In communications, the portion of a Data Terminal Equipment (DTE) device that sends data.

data stream *n.* An undifferentiated, byte-by-byte flow of data.

data structure *n.* An organizational scheme, such as a record or array, that can be applied to data to facilitate interpreting the data or performing operations on it.

data switch *n.* A device in a computer system that routes incoming data to various locations.

Data Terminal Equipment *n. See* DTE.

Data Terminal Ready *n. See* DTR.

data traffic *n.* The exchange of electronic messages—control and data—across a network. Traffic capacity is measured in bandwidth; traffic speed is measured in bits per unit of time.

data transfer *n.* The movement of information from one location to another, either within a computer (as from a disk drive to memory), between a computer and an external device (as between a file server and a computer on a network), or between separate computers.

data transfer rate *n. See* data rate.

data transmission *n.* The electronic transfer of information from a sending device to a receiving device.

data type *n.* In programming, a definition of a set of data that specifies the possible range of values of the set, the operations that can be performed on the values, and the way in which the values are stored in memory. Defining the data type allows a computer to manipulate the data appropriately. Data types are most often supported in high-level languages and often include types such as real, integer, floating point, character, Boolean, and pointer. How a language handles data typing is one of its major characteristics. *See also* cast, constant, enumerated data type, strong typing, type checking, user-defined data type, variable, weak typing.

data validation *n.* The process of testing the accuracy of data.

data value *n.* The literal or interpreted meaning of a data item, such as an entry in a database, or a type, such as an integer, that can be used for a variable.

data warehouse[1] *n.* A database, frequently very large, that can access all of a company's information. While the warehouse can be distributed over several computers and may contain several databases and information from numerous sources in a variety of formats, it should be accessible through a server. Thus, access to the warehouse is transparent to the user, who can use simple commands to retrieve and analyze all the information. The data warehouse also contains data about how the warehouse is organized, where the information can be found, and any connections between data. Frequently used for decision support within an organization, the data warehouse also allows the organization to organize its data, coordinate updates, and see relationships between information gathered from different parts of the organization. *See also* database, decision support system, server (definition 1), transparent (definition 1).

data warehouse[2] *vb.* To acquire, collect, manage, and disseminate information gathered from various sources into a single location; or to implement an informational database used to store sharable data. Data warehousing is a four-step process: gathering data; managing the data in a centralized location; providing access to the data along with tools for interpreting, analyzing, and reporting on the data; and producing reports on the data to be used for decision making. *See also* downflow, inflow, metaflow, upflow.

date and time stamp *n. See* time stamp.

date counter overflow *n.* A problem that may occur in systems or programs when the value in a date variable exceeds allowable values. A date counter overflow can occur when an incremental date produces a number that the system interprets as zero or a negative number. This is likely to cause the system or program to post an error message in turn or to revert to the original starting point. Although this was largely considered a Year 2000 problem, such an error is not necessarily confined to the year 2000.

date dependency *n.* In terms of the Year 2000 problem, the need many programs have for date-related input or output data and the way dates are represented in that data. This dependency affects whether the program can run correctly when the turn of the century is reached.

COBOL *n.* Acronym for Common Business-Oriented Language. A verbose, English-like compiled programming language developed between 1959 and 1961 and still in widespread use today, especially in business applications typically run on mainframes. A COBOL program consists of an Identification Division, which specifies the name of the program and contains any other documentation the programmer wants to add; an Environment Division, which specifies the computers being used and the files used in the program for input and output; a Data Division, which describes the format of the data structures used in the program; and a Procedure Division, which contains the procedures that dictate the actions of the program. *See also* compiled language.

cobweb site *n.* A Web site that is far out of date. *See also* Web site.

Cocoa *n.* A set of object-oriented development tools and interfaces available on Mac OS X. Cocoa contains a set of frameworks, software components, and development tools used to construct applications for Mac OS X and provides programming interfaces in Java and Objective-C. Cocoa is based on NeXT's OpenStep and is integrated with Apple technologies.

CODASYL *n.* Acronym for Conference on Data Systems Languages. An organization founded by the U.S. Department of Defense. CODASYL is dedicated to the development of data-management systems and languages, among them the widely used COBOL.

code¹ *n.* **1.** Program instructions. Source code consists of human-readable statements written by a programmer in a programming language. Machine code consists of numerical instructions that the computer can recognize and execute and that were converted from source code. *See also* data, program. **2.** A system of symbols used to convert information from one form to another. A code for converting information in order to conceal it is often called a *cipher*. **3.** One of a set of symbols used to represent information.

code² *vb.* To write program instructions in a programming language. *See also* program.

code access security *n.* A mechanism provided by the runtime whereby managed code is granted permissions by security policy and these permissions are enforced, limiting what operations the code will be allowed to perform. To prevent unintended code paths from exposing a security vulnerability, all callers on the call stack must be granted the necessary permissions (possibly subject to override by assertion or denial).

codec *n.* **1.** Short for coder/decoder. Hardware that can convert audio or video signals between analog and digital forms. **2.** Short for compressor/decompressor. Hardware or software that can compress and uncompress audio or video data. *See also* compress², uncompress. **3.** Hardware that combines the functions of definitions 1 and 2.

code conversion *n.* **1.** The process of translating program instructions from one form into another. Code may be converted at the source-language level (for example, from C to Pascal), at the hardware-platform level (for example, from working on the IBM PC to working on the Apple Macintosh), or at the language level (for example, from source code in C to machine code). *See also* code¹ (definition 1). **2.** The process of transforming data from one representation to another, such as from ASCII to EBCDIC or from two's complement to binary-coded decimal.

Code Division Multiple Access *n.* A form of multiplexing in which the transmitter encodes the signal, using a pseudo-random sequence that the receiver also knows and can use to decode the received signal. Each different random sequence corresponds to a different communication channel. Motorola uses Code Division Multiple Access for digital cellular phones. *Acronym:* CDMA. *Also called:* spread spectrum. *See also* multiplexing, transmitter.

code page *n.* In MS-DOS versions 3.3 and later, a table that relates the binary character codes used by a program to keys on the keyboard or to the appearance of characters on the display. Code pages are a means of providing support for character sets and keyboard layouts used in different countries. Devices such as the display and the keyboard can be configured to use a specific code page and to switch from one code page (such as United States) to another (such as Portugal) at the user's request.

code profiler *n.* A tool designed to aid developers in identifying and eliminating the code inefficiencies that cause bottlenecks and degrade performance in their applications. Code profilers analyze an executing application to determine both how long functions take to execute and how often they are called. Using a code profiler is a repetitive process in that the tool must be reused after each section of inefficient code has been found and corrected.

coder *n.* *See* programmer.

Code Red worm *n.* A fast-spreading and pernicious Internet worm first discovered in mid-2001. The Code Red

propagates
once is po
Code Red worm is
mode from th
in flood mode
in hibernati
when the cycle beg
all computers prey
will be attacked ea
machine. This mak
cult because a sing
earlier propagation
every machine on t
subject to multiple
Code Red worm ar
was named for a ca
team that first track

code segment *n.*
gram instructions.
program's code typ
operations. Code s
into memory as me
segment is kept in
loaded only when t

code signing *n.* T
to additions and up
tions published on
to provide a level o
distribution. *See al*

code snippet *n.* 1
ming instructions e
defined by the user
more lines of sourc
button does when c
programming code
ally the code snipp

coding form *n.* A
vertical lines to aid
guages that have p
TRAN). Most prog
use paper at all.

coercion *n.* *See* ca

Coffee Pot Contr

coherence *n.* **1.** Ir
of the value of one

nications. The addition of SBC's Internet customer base made Prodigy the third largest ISP in the United States.

Prodigy Information Service *n.* An online information service founded by IBM and Sears. Like its competitors America Online and CompuServe, Prodigy offers access to databases and file libraries, online chat, special interest groups, e-mail, and Internet connectivity. *Also called:* Prodigy.

product *n.* **1.** An operator in the relational algebra used in database management that, when applied to two existing relations (tables), results in the creation of a new table containing all possible ordered concatenations (combinations) of tuples (rows) from the first relation with tuples from the second. The number of rows in the resulting relation is the product of the number of rows in the two source relations. *Also called:* Cartesian product. *Compare* inner join. **2.** In mathematics, the result of multiplying two or more numbers. **3.** In the most general sense, an entity conceived and developed for the purpose of competing in a commercial market. Although computers are products, the term is more commonly applied to software, peripherals, and accessories in the computing arena.

production system *n.* In expert systems, an approach to problem solving based on an "IF this, THEN that" approach that uses a set of rules, a database of information, and a "rule interpreter" to match premises with facts and form a conclusion. Production systems are also known as rule-based systems or inference systems. *See also* expert system.

Professional Graphics Adapter *n.* A video adapter introduced by IBM, primarily for CAD applications. The Professional Graphics Adapter is capable of displaying 256 colors, with a horizontal resolution of 640 pixels and a vertical resolution of 480 pixels. *Acronym:* PGA.

Professional Graphics Display *n.* An analog display introduced by IBM, intended for use with their Professional Graphics Adapter. *See also* Professional Graphics Adapter.

profile¹ *n. See* user profile.

profile² *vb.* To analyze a program to determine how much time is spent in different parts of the program during execution.

profiler *n.* A diagnostic tool for analyzing the run-time behavior of programs.

Profiles for Open Systems Internetworking Technology *n. See* POSIT.

program¹ *n.* A sequence of instructions that can be executed by a computer. The term can refer to the original source code or to the executable (machine language) version. *Also called:* software. *See also* program creation, routine, statement.

program² *vb.* To create a computer program, a set of instructions that a computer or other device executes to perform a series of actions or a particular type of work.

program button *n.* On a handheld device, a navigation control that is pressed to launch an application. *Also called:* application button.

program card *n. See* PC Card, ROM card.

program cartridge *n. See* ROM cartridge.

program comprehension tool *n.* A software engineering tool that facilitates the process of understanding the structure and/or functionality of computer applications. *Acronym:* PCT. *Also called:* software exploration tool.

program counter *n.* A register (small, high-speed memory circuit within a microprocessor) that contains the address (location) of the instruction to be executed next in the program sequence.

program creation *n.* The process of producing an executable file. Traditionally, program creation comprises three steps: (1) compiling the high-level source code into assembly language source code; (2) assembling the assembly language source code into machine-code object files; and (3) linking the machine-code object files with various data files, run-time files, and library files into an executable file. Some compilers go directly from high-level source to machine-code object, and some integrated development environments compress all three steps into a single command. *See also* assembler, compiler (definition 2), linker, program.

program encapsulation *n.* A method of dealing with programs with Year 2000 problems that entailed modifying the data with which a program worked. The input data is modified to reflect a parallel date in the past that the program can handle. When output is generated, that data is changed again, to reflect the correct date. The program itself remains unchanged.

program file *n.* A disk file that contains the executable portions of a computer program. Depending on its size and

internet.com (Webopedia) The #1 online encyclopedia dedicated to computer technology

transcoding

Last modified: Friday, October 24, 2008

The process of converting a media file or object from one format to another. Transcoding is often used to convert video formats (i.e., Beta to VHS, VHS to QuickTime, QuickTime to MPEG). But it is also used to fit HTML files and graphics files to the unique constraints of mobile devices and other Web-enabled products. These devices usually have smaller screen sizes, lower memory, and slower bandwidth rates. In this scenario, transcoding is performed by a transcoding proxy server or device, which receives the requested document or file and uses a specified annotation to adapt it to the client.